



UNITED STATES PATENT AND TRADEMARK OFFICE

NN

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/628,960	07/28/2003	Carlos Bonilla	200309109-1	6175
22879	7590	04/30/2007		
HEWLETT PACKARD COMPANY P O BOX 272400, 3404 E. HARMONY ROAD INTELLECTUAL PROPERTY ADMINISTRATION FORT COLLINS, CO 80527-2400			EXAMINER	
			DAO, THUY CHAN	
			ART UNIT	PAPER NUMBER
			2192	
			MAIL DATE	DELIVERY MODE
			04/30/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/628,960	BONILLA, CARLOS
	Examiner	Art Unit
	Thuy Dao	2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 06 February 2007.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) _____ is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-20 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 28 July 2003 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____.
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date _____.
- 5) Notice of Informal Patent Application
- 6) Other: _____.

DETAILED ACTION

1. This action is responsive to the amendment filed on February 6, 2007.
2. Claims 1-20 have been examined.

Response to Amendments

3. Per Applicant's request, claims 5-8 and 10-20 have been amended.
4. The objection to drawings is withdrawn in view of Applicant's amendments.
5. The objection to the specification is withdrawn in view of Applicant's amendments.
6. The objection to claims 5-20 is withdrawn in view of Applicant's amendments.

Response to Arguments

7. The Applicant is thanked for a thorough reply. Applicant's arguments have been fully considered. However, they are not persuasive.

a) The Applicant asserted, "...*the present rejection picks and chooses information from a variety of locations and embodiments/examples within the Evans reference in an attempt to assemble the elements of the invention as claimed*..." (Remarks, page 11: 24-26, page 12: 25 – page 13: 2, and page 13: 21-25).

The examiner respectfully disagrees. As clearly indicated in col.4: 15-20, features of ICAT ("Interactive Code Analysis Tool", a existing debugger from International Business Machines IBM) are incorporated into ICAT2 (Evans' teaching).

ICAT "is used as a starting point, with modifications as disclosed herein in order to migrate the existing ICAT to support JPDA", wherein JPDA as Java Platform Debugger Architecture (see col.4: 23-31).

For this reason, Evans' teaching (new features in ICAT2 and existing features of ICAT, which actually means existing features of ICAT2) directs to a single method and system which

"... *provide improved source-level debugging capabilities of an object-oriented application program which may include linked native language dynamic load libraries ...*" (emphasis added, see col.3, Summary of the invention).

Accordingly, Applicant's arguments are not persuasive.

b) Claims 1-4 (Remarks, pp. 11-12):

Evans discloses *an emulation and native language interface test method comprising:*

initializing an emulation language virtual machine (e.g., FIG. 1, initializing JVM 16, which is running the ICAT2 probe 41, col.5: 7-12);

wrapping native language code in a simulation test macro which creates simulated interfacing problems (e.g., col.1: 7-11, debugging a Java application that includes native method dynamic load libraries; col.2: 15-22, a first "daemon" process that performs native method debugging); and

examining reaction to said simulated interfacing problems when an emulation language application is run (e.g., FIG. 6, Interface Code Analysis Tool 41 (ICAT2) with event handlers 61-67 (examining reaction to said simulated interfacing problems); GUI of ICAT, ICAT2 to set breakpoints, step applications, examine application stack and variables, col.2: 33-47).

Accordingly, Applicant's arguments are not persuasive. The examiner respectfully maintains ground of rejection over claims 1-4.

c) Claims 5-8 (Remarks, pp. 12-13):

Evans discloses *a Java Native Language Interface testing system comprising:*

means for communicating information (e.g., FIG. 6, Interface Code Analysis Tool 41 (ICAT2) with event handlers 61-67; GUI of ICAT, ICAT2 to set breakpoints, step applications, examine application stack and variables, col.2: 33-47);

means for processing said information, including instructions for testing a Java Native Language Interface, said means for processing said information coupled to said means for communicating information (e.g., col.1: 7-11; col.2: 15-22); and

means for storing said information, including said instructions for testing said Java Native Language Interface, said means for storing said information coupled to

Art Unit: 2192

said means for communicating information (e.g., col.6: 52-57; col.8: 15-19; col.8: 25- col.9: 67; FIG. 13, data sent to application, data output from application).

Accordingly, Applicant's arguments are not persuasive. The examiner respectfully maintains ground of rejection over claims 5-8.

d) Claims 9-20 (Remarks, pp. 13-14):

Evans discloses a *Java Native Language Interface test method comprising:*

investigating Java Native Language Interface test mode status (e.g., col.5: 62 – col.6: 5, if debugging is enabled, calling “launh_then_attach”, passing arguments to each connector, ..., setting “suspend” argument to true);

running a Java application with simulated Java Native Language Interface problems if said Java Native Language Interface test mode is enabled (e.g., FIG. 1, col.5: 7-12; col.1: 7-11; col.2: 15-22); and

initiating a call to a Java Native Language Interface function directly without said simulated Java Native Language Interface problems if said Java Native Language Interface test mode is not enabled (e.g., col.1: 32-37, Java application with native method DLL's, which calls native method dynamic load libraries directly when executing; col.1: 66 – col.2: 5, said application executes with debugging enabled or not enabled).

Accordingly, Applicant's arguments are not persuasive. The examiner respectfully maintains ground of rejection over claims 9-20.

Claim Rejections – 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States

only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

9. Claims 1-20 are rejected under 35 U.S.C. 102(e) as being anticipated by Evans (art of record, US Patent No. 6,826,746).

Claim 1:

Evans discloses *an emulation and native language interface test method comprising:*

initializing an emulation language virtual machine (e.g., FIG. 1, initializing JVM 16, which is running the ICAT2 probe 41, col.5: 7-12);

wrapping native language code in a simulation test macro which creates simulated interfacing problems (e.g., col.1: 7-11, debugging a Java application that includes native method dynamic load libraries; col.2: 15-22, a first “daemon” process that performs native method debugging); and

examining reaction to said simulated interfacing problems when an emulation language application is run (e.g., FIG. 6, Interface Code Analysis Tool 41 (ICAT2) with event handlers 61-67 (examining reaction to said simulated interfacing problems); GUI of ICAT, ICAT2 to set breakpoints, step applications, examine application stack and variables, col.2: 33-47).

Claim 2:

The rejection of claim 1 is incorporated. Evans also discloses *said emulation language virtual machine creates a runtime environment and said runtime environment can include a class loader subsystem and an execution engine* (e.g., FIG. 1, class loader subsystem in JVM 16, engine 13, col.5: 2-17).

Claim 3:

The rejection of claim 1 is incorporated. Evans also discloses *said simulated test problems include simulations of error conditions associated with a native language code method attempt to respond to a call from emulation language code* (e.g., FIG. 6, ICAT2

with event handlers 61-67; GUI of ICAT, ICAT2 to set breakpoints, step applications, examine application stack and variables, col.2: 33-47).

Claim 4:

The rejection of claim 1 is incorporated. Evans also discloses *forwarding an indication that there is a insufficient memory allocation exception to a native language method attempting to ascertain an indication of a memory location for information associated with a native language function* (e.g., col.8: 1-8, Memory Usage Information).

Claim 5:

Evans discloses a *Java Native Language Interface testing system comprising:*
means for communicating information (e.g., FIG. 6, Interface Code Analysis Tool 41 (ICAT2) with event handlers 61-67; GUI of ICAT, ICAT2 to set breakpoints, step applications, examine application stack and variables, col.2: 33-47);
means for processing said information, including instructions for testing a Java Native Language Interface, said means for processing said information coupled to said means for communicating information (e.g., col.1: 7-11; col.2: 15-22); and
means for storing said information, including said instructions for testing said Java Native Language Interface, said means for storing said information coupled to said means for communicating information (e.g., col.6: 52-57; col.8: 15-19; col.8: 25- col.9: 67; FIG. 13, data sent to application, data output from application).

Claim 6:

The rejection of claim 5 is incorporated. Evans also discloses *said means for processing performs a Java Native Language Interface test method* (e.g., FIG. 1, ICAT2 Probe 41, col.5: 2-12).

Claim 7:

The rejection of claim 5 is incorporated. Evans also discloses *an interface testing macro module* (e.g., FIG. 1, JDaemon DLL 14, col. 5: 5-8; col.2: 15-22)

Claim 8:

The rejection of claim 5 is incorporated. Evans also discloses *emulating a Java virtual machine* (e.g., FIG. 15, JVM 16, col.4: 40-51).

Claim 9:

Evans discloses a *Java Native Language Interface test method comprising:*

investigating Java Native Language Interface test mode status (e.g., col.5: 62 – col.6: 5, if debugging is enabled, calling “launh_then_attach”, passing arguments to each connector, ..., setting “suspend” argument to true);

running a Java application with simulated Java Native Language Interface problems if said Java Native Language Interface test mode is enabled (e.g., FIG. 1, col.5: 7-12; col.1: 7-11; col.2: 15-22); and

initiating a call to a Java Native Language Interface function directly without said simulated Java Native Language Interface problems if said Java Native Language Interface test mode is not enabled (e.g., col.1: 32-37, Java application with native method DLL’s, which calls native method dynamic load libraries directly when executing; col.1: 66 – col.2: 5, said application executes with debugging enabled or not enabled).

Claim 10:

The rejection of claim 9 is incorporated. Evans also discloses *said Java Native Language Interface test mode status indicator indicates if said Java Native Language Interface test mode status is enabled* (e.g., FIG. 1, JVM 16, col.5: 7-12; col.1: 7-11; col.2: 15-22).

Claim 11:

The rejection of claim 9 is incorporated. Evans also discloses *said Java Native Language Interface test mode status indicator is a flag wherein a state of said flag indicates if said Java Native Language Interface test mode status is set* (e.g., col.5: 62 – col.6: 5).

Claim 12:

The rejection of claim 9 is incorporated. Evans also discloses *a register value indicates said Java Native Language Interface test mode status* (e.g., col.5: 62 – col.6: 5).

Claim 13:

The rejection of claim 9 is incorporated. Evans also discloses *identifying indications of Java Native Language Interface code trouble associated with out of memory situations* (e.g., col.8: 1-8, Memory Usage Information).

Claim 14:

The rejection of claim 9 is incorporated. Evans also discloses *a Java Native Language Interface problem simulation process is performed to simulate Java Native Language Interface problems* (e.g., col.4: 40-51).

Claim 15:

The rejection of claim 9 is incorporated. Evans also discloses:

determining a Java Native Language Interface problem simulation occurrence level (e.g., col.7: 52-63);

introducing simulation randomness (e.g., col.6: 10-21);

performing an analysis whether to initiate a simulation of Java Native Language Interface problem; calling a Java Native Language Interface memory allocation function normally (e.g., col.8: 2-8);

forwarding a Java Native Language Interface problem indicator automatically; and implementing a reaction to the Java Native Language Interface problem indication (e.g., col.8: 63 – col.9: 14).

Claim 16:

The rejection of intervening claim 15 is incorporated. Evans also discloses:

looking up a predefined Java Native Language Interface problem simulation occurrence level (e.g., col.8: 2-8);

generating a random value (e.g., col.6: 10-21); and

correlating said random value to said JNI problem simulation occurrence level (e.g., col.7: 52-63).

Claim 17:

The rejection of intervening claim 16 is incorporated. Evans also discloses:

comparing said randomly generated value to said Java Native Language Interface problem simulation occurrence level (e.g., col.6: 10-21); and

initiating a simulation of a Java Native Language Interface problem if said generated value from is less than said Java Native Language Interface problem simulation occurrence level (e.g., col.8: 63 – col.9: 14).

Claim 18:

The rejection of intervening claim 16 is incorporated. Evans also discloses *initiating a controlled shut down (e.g., col.5; 62 – col.6: 5).*

Claim 19:

The rejection of intervening claim 16 is incorporated. Evans also discloses *clearing a system and canceling information inventory collections that is occupying memory space (e.g., col.8: 2-8).*

Claim 20:

The rejection of intervening claim 16 is incorporated. Evans also discloses *providing an indication of the Java Native Language Interface problem to a user* (e.g., col.8: 63 – col.9: 14).

Conclusion

10. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone is (571) 272 8570. The examiner can normally be reached on the first Monday of the bi-week, and every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR.

Art Unit: 2192

Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

T. Dao



WEI ZHEN
SUPERVISORY PATENT EXAMINER